Politecnico di Torino

Digital systems electronics
A.A. 2018/2019

Prof. G. Masera

Lab 08

28 may 2019

Berchialla Luca	236032
Laurasi Gjergji	238259
Mattei Andrea	233755
Lombardo Domenico Maria	233959
Wylezek Karolina	267219

Output compare VS Reload

In the following report the Output Compare register and/or auto-reload register can be calculated from the desired asserted flag frequency using the following formula:

$$F_X = \frac{F_{CLK}}{(OC+1)\cdot(ARR+1)} \qquad (1)$$

Notice that to correctly generate a square wave with a desired frequency F_X exploiting the OC or ARR approach, the formula 1 should also be divided by a 2 factor.

1.1

Following the instructions on the lab8 document we created the project with cube mx, where just had to set the auto reload register to the proper value calculated with the formula mentioned before.
In the source code we polled the counter register to a random integer smaller or equal than the ARR value to toggle the output pin (in our code we used the arr value).
With the oscilloscope we can clearly see that this approach doesn't work because just a few times the MCU is detecting that the counter register and our random integer are equal. Instead when we use the condition "$counter <= arr$" the pin is toggled every time that the MCU executes it, therefor we observed a frequency of 1.6 MHz.

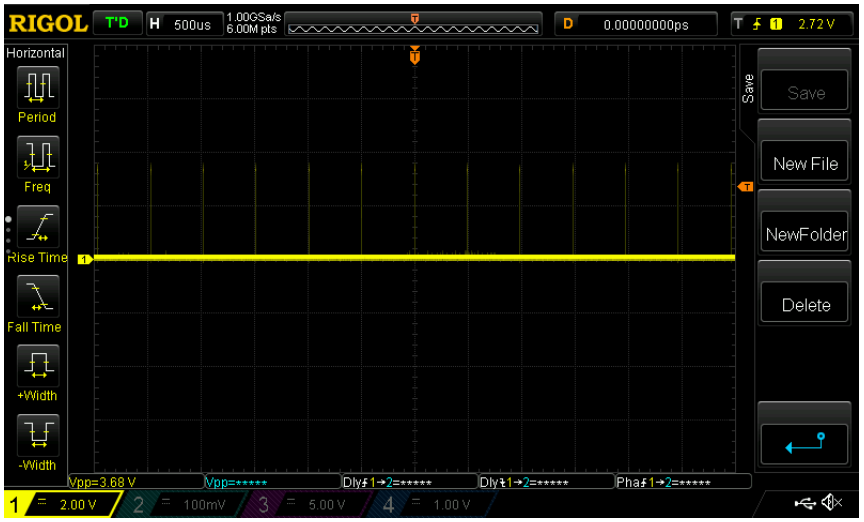

Figure 1: condition "$counter = arr$"

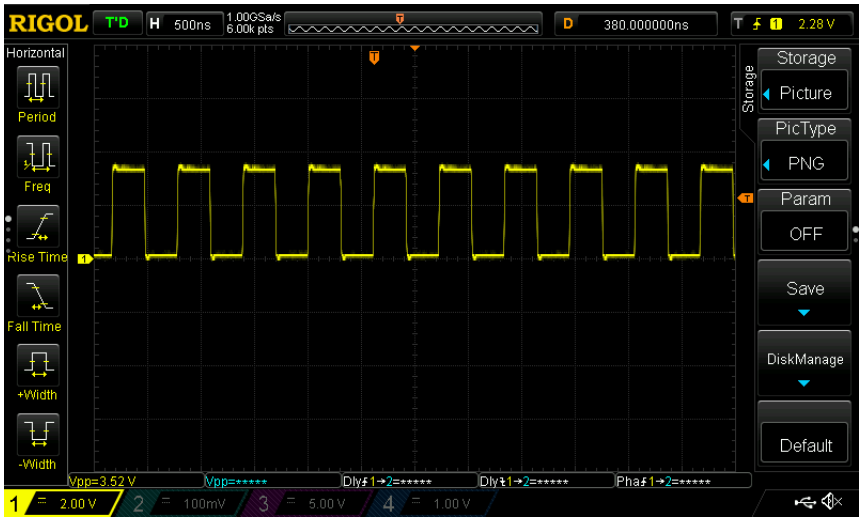

Figure 2: condition "$counter <= arr$"

1.2

In this section we reused the code of the previous point but this time we polled the auto reload register flag.

If we try to achieve 4 MHz we can see that the board can't deliver such a high

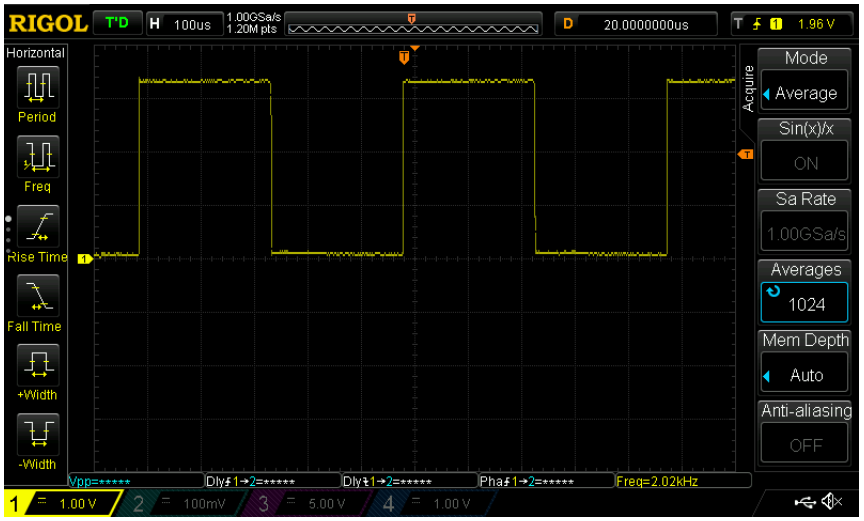

Figure 3: 2 kHz wave

frequency, actually we registered 1.6 MHz.

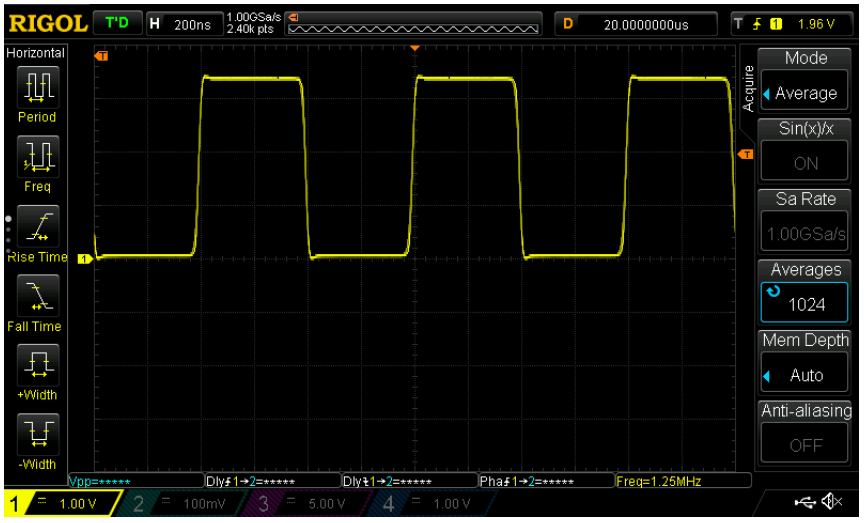

Figure 4: should be 4Mhz wave

1.3

In this section we had to use the capture compare unit in order to generate the square wave, to use this approach we have to let the counter count up to its maximum value. To generate the wave we use the flag generated by the capture compare unit. The capture compare register is incremented every time we detect the flag in order to have an event every half period. The value of the increment is calculated with the formula mentioned before.

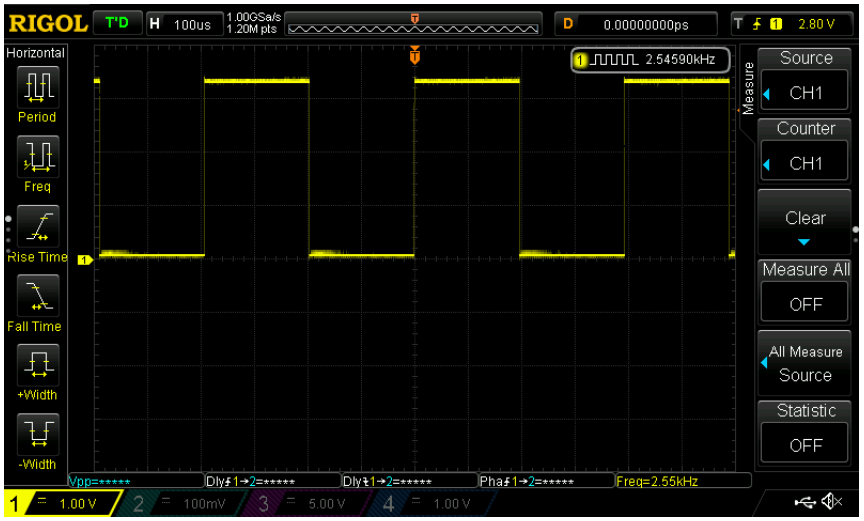

Figure 5: should be 4Mhz wave

1.4

In this section we had to generate a square wave with variable frequency between 800Hz and 4.5kHz. To control the output frequency it was used a potentiometer to generate a variable voltage. Than we sample this voltage with the ADC1 peripheral in continuous mode and in the software we poll the *EOC* flag to know when a new sample is ready to be read. With the value we read we can generate a variable increment to be added to the capture compare register in order to obtain a variable output frequency between 800Hz and 4.5kHz.

Here we report the value of the increment of the oc register for each output frequency. We can observe that for increasing frequencies the increment becomes smaller thus generating an event more often.

800 Hz $->$increment of oc register: 531
1.2 kHz $->$ increment of oc register: 353
3.0 kHz $->$ increment of oc register: 142
4.5 kHz $->$ increment of oc register: 95